LESS IS MORE: SELECTIVE EXPERT SUPERVISION FOR MULTI-TEACHER ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain-specialist language-model agents can perform well individually, but integrating their capabilities into one student remains difficult. Multi-Teacher On-Policy Distillation (MOPD) addresses sequential-training interference by supervising student rollouts with domain specialists, yet our four-domain study reveals uneven transfer: WebShop success is 57.29% for its teacher and only 40.53% for the Vanilla MOPD student. A directed gradient analysis of Vanilla MOPD further finds time-varying opposition among domain learning signals, raising the question of whether every student trajectory needs full teacher supervision. We propose SEAM (Selective Expert Supervision and Advantage Matching), which preserves rollout quotas and fixed teacher routing but scores every failed trajectory and only a sampled subset of successful ones. To limit changes in domain influence when retained batches shrink, SEAM calibrates each domain's distillation signal to its original rollout budget before a shared student update. Across ALFWorld, WebShop, SciWorld, and Search, SEAM improves WebShop success by 9.35 percentage points and four-domain macro success by 2.20 points over Vanilla MOPD across two training seeds. It uses 5.75% fewer teacher-scored trajectories and reduces measured end-to-end training time by 5.48%. Selective supervision thus improves integration without reducing exploration, although specialist performance is not inherited losslessly.

1 INTRODUCTION

Language-model agents have made substantial progress in interacting with complex environments. ReAct showed that interleaving reasoning and actions improves performance on interactive tasks such as ALFWorld and WebShop; SWE-agent extended agentic interaction to repository navigation, code editing, and testing; and subsequent reinforcement-learning methods such as GiGPO have improved long-horizon agent training through finer-grained credit assignment (Yao et al., 2023; Yang et al., 2024; Feng et al., 2025). Together, these advances make it increasingly practical to train strong agents for individual domains—and increasingly important to integrate their capabilities into a single agent.

Capability integration remains difficult. When reinforcement learning is applied sequentially across domains, gains in a newly trained domain can coincide with degradation in previously acquired capabilities, creating a see-saw effect (Ma et al., 2026). Multi-Teacher On-Policy Distillation (MOPD) addresses this problem by training domain-specialist RL teachers independently, then distilling them into one student on the student's own trajectories. Each trajectory is scored by its corresponding domain teacher, providing dense token-level supervision without sequentially retraining the student for each capability (Ma et al., 2026).

Yet MOPD does not always inherit every specialist's capability equally well. We study ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2022), SciWorld (ScienceWorld) (Wang et al., 2022), and a Search environment for tool-assisted question answering related to Search-R1 (Jin et al., 2025). In this four-domain Qwen2.5-3B agent setting, Figure 1 (left) shows a pronounced gap on WebShop: the specialist teacher achieves 57.29% success, whereas the MOPD student achieves 40.53%, a difference of 16.76 percentage points. The corresponding gaps are much smaller on ALFWorld (2.97 points), SciWorld (0.55 points), and Search (0.11 points).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

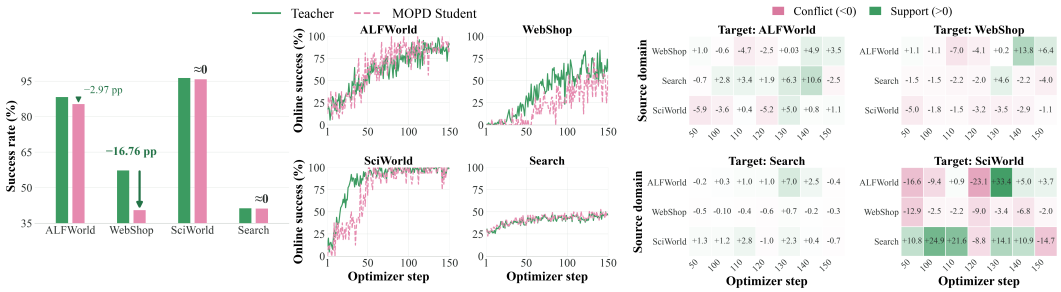

Figure 1: Capability integration and gradient interactions in four agent domains. Left: mean held-out success rates from three evaluations per specialist Teacher and eight evaluations per Vanilla MOPD Student checkpoint, averaged across two student-training runs. Middle: online training success curves. Right: directed pairwise first-order OPD gradient effects, expressed relative to the target domain's own descent term; negative values indicate opposition.

To investigate this imbalance, we examine the online training-success trajectories in Figure 1 (middle). The MOPD student's WebShop curve stays below the independently trained WebShop teacher's curve for most of training, whereas the trajectories in the other domains intersect or remain much closer. MOPD routes each trajectory to its matching specialist teacher, so a teacher's behavior on *other* domains is not directly used as supervision. Nevertheless, all four domains update the same student parameters. This raises a more specific question: can the learning signal from one domain oppose the learning signal needed by another?

Gradient conflict is a recognized difficulty in multi-task optimization (Yu et al., 2020). We probe its role here through a full-parameter, directed pairwise gradient analysis of Vanilla MOPD at seven optimizer updates after the first step. For each update, we compute the domain-specific OPD gradients on the current batch and measure each source domain's first-order contribution relative to the target domain's own descent term. Figure 1 (right) reveals time-varying interference: at update 110, ALFWorld, Search, and SciWorld oppose WebShop's own descent by approximately 7.0%, 2.2%, and 1.5%, respectively, for a combined offset of 10.68%; the combined offset is 9.24% at update 120. Other inspected updates instead show net support. Thus, cross-domain opposition exists but is not persistent. These findings raise a question: if training signals from different domains can conflict, are all training samples necessary for effective capability integration?

Motivated by this observation, we introduce **SEAM** (**S**elective **E**xpert Supervision and **A**dvantage **M**atching), an outcome-aware MOPD variant that preserves the student rollout budget and fixed domain-to-teacher routing. Failures expose behavior that has not yet succeeded, whereas scoring every successful rollout can spend teacher and actor computation on behavior the student already performs correctly. SEAM therefore retains every failure but samples a limited number of successes, keeping positive examples rather than discarding them altogether. Selection also changes how many trajectories each domain contributes to a shared update; without a correction, thinning one domain could inadvertently change its influence relative to the others. SEAM consequently calibrates each retained domain's OPD signal against its original rollout budget before combining the domains in one student update.

Across ALFWorld, WebShop, SciWorld, and Search, SEAM achieves mean success rates of 83.94%, 49.88%, 96.95%, and 40.84% over two student-training seeds, with each final checkpoint evaluated eight times per domain. Relative to Vanilla MOPD, it improves WebShop by 9.35 percentage points, SciWorld by 1.15 points, and four-domain macro success by 2.20 points, closing 55.8% of the WebShop teacher–student gap displayed in Figure 1; ALFWorld and Search decrease by 1.38 and 0.33 points. SEAM reduces teacher-scored trajectories by 5.75% while preserving the full student rollout budget; end-to-end training time is 5.48% shorter in the measured run. These results show that full teacher scoring and actor training on every student rollout are not necessary for improved aggregate capability integration: selective supervision coupled with signal calibration can achieve better results with less downstream training work.

Our contributions are threefold:

1. We identify an uneven capability-transfer failure in four-domain agentic MOPD, challenging the assumption that distilling independently strong specialists preserves each of their strengths. WebShop has the largest teacher–student gap, and directed full-parameter gradient measurements at seven training updates show that other domains can oppose its OPD descent, with the effect changing over time. This observation motivates examining which trajectories need supervision rather than suppressing a domain throughout training.

2. We develop SEAM to test whether every generated student trajectory must undergo teacher scoring and actor optimization. It keeps the on-policy rollout quotas and fixed domain specialists, retains every failure and a limited number of successes before teacher scoring, and calibrates each retained domain against its original rollout budget in a balanced shared update. The design preserves feedback on unresolved behavior while limiting the risk that a small batch from a nearly solved domain acquires disproportionate influence.

3. We demonstrate targeted capability recovery with less downstream training work: across two student-training seeds, SEAM improves WebShop success by 9.35 percentage points and four-domain macro success by 2.20 points over Vanilla MOPD, while using 5.75% fewer teacher-scored trajectories and 5.48% less end-to-end time in the measured run. The remaining teacher–student gap makes clear that this is an improvement in integration, not lossless inheritance.

2 Related Work

LLM Agent Training. Interactive agents are trained and evaluated in embodied, web-shopping, scientific, and search environments (Shridhar et al., 2021; Yao et al., 2022; Wang et al., 2022; Jin et al., 2025). Online RL addresses sparse outcomes and policy drift through curricula and stabilized multi-turn interaction (Qi et al., 2025; Wang et al., 2025; Xi et al., 2025), while finer credit assignment and evidence calibration refine how delayed outcomes shape decisions (Feng et al., 2025; Zheng et al., 2026; Liu et al., 2026; Li et al., 2026c). A complementary direction internalizes skills extracted from successful interaction (He et al., 2026b), and scalable systems separate environment execution from policy optimization (Luo et al., 2025). On-policy teacher feedback also trains agents on their own rollouts (Agarwal et al., 2024; Li et al., 2026a). These methods advance capability acquisition or agent training; SEAM asks which already-generated multi-domain trajectories need teacher scoring and a shared actor update.

On-Policy Distillation (OPD). OPD supervises a student's own samples with teacher probabilities, reducing the state mismatch of static demonstrations (Agarwal et al., 2024; Ye et al., 2026). Its behavior depends on teacher–student compatibility and can change sampling efficiency without necessarily expanding the student's solution set (Li et al., 2026b; Ge et al., 2026). Other work studies parameter-update efficiency (Cai et al., 2026), guides early agent turns (Li et al., 2026a), stops or restarts unreliable trajectories (Gui et al., 2026), or retires teacher supervision as an agent matures (Yu et al., 2026). Controlled planning studies further expose when OPD helps or conflicts with prior capabilities (Men et al., 2026). Reward-aligned filtering uses the teacher-derived signal and outcome feedback to remove misaligned updates (Gan et al., 2026). SEAM makes a different, pre-scoring decision: with fixed domain specialists and unchanged rollout quotas, it retains every failure, subsamples successes, and calibrates the retained signals within one multi-domain student update.

Multi-Teacher On-Policy Distillation (MOPD). MOPD integrates independently trained specialists by having domain-matched teachers score the student's own trajectories (Ma et al., 2026). Subsequent methods address imbalance by adjusting token-level domain budgets and refreshing stale feedback (Gao et al., 2026), or by changing how many rollouts each domain generates (Sun et al., 2026b). Others revise which teacher supplies supervision for a sample or token, using answer verification or local reliability (He et al., 2026a; Sun et al., 2026a; Xu & Zhang, 2026). SEAM addresses a different decision: which *already generated student trajectories* need teacher scoring and actor training? It keeps the rollout quota and domain-matched teacher fixed, retains every failure while subsampling successes within each domain before teacher scoring, and matches the retained OPD signal to the original domain budget. Thus, unlike domain rescheduling or teacher selection, SEAM changes trajectory admission without changing the token-level OPD advantage.

3 METHODOLOGY

Figure 1 shows that domains can exert opposing effects on a shared MOPD update, but the sign changes over training; it does not identify a fixed set of harmful samples. We therefore ask a narrower question: after the student has generated a multi-domain batch, does every trajectory need teacher scoring and an actor update? SEAM tests this question at the boundary between rollout generation and distillation. It preserves the number of student rollouts and the domain-to-teacher assignment, while changing which completed trajectories proceed to the expensive downstream stages.

MOPD Baseline. Let $\mathcal{D} = \{\text{ALFWorld}, \text{WebShop}, \text{SciWorld}, \text{Search}\}$, and let π_{θ_0} be the common initial policy. As in MOPD (Ma et al., 2026), we first obtain one specialist π_{ϕ_d} per domain, freeze the specialists, and initialize a shared student π_θ from π_{θ_0}. In this study, the four specialists are independently trained with Group-in-Group Policy Optimization (GiGPO) (Feng et al., 2025); both Vanilla MOPD and SEAM use these same fixed teachers. At each student update, domain d contributes B_d on-policy trajectories $\tau_{d,i}$; the quotas are $(16, 16, 16, 256)$ in the domain order above. A trajectory contains environment observations and student-generated response tokens. Let $h_{d,i,t}$ be the complete interaction prefix before sampled token $a_{d,i,t}$, and let $m_{d,i,t} \in \{0, 1\}$ mask valid response tokens. Vanilla MOPD dispatches *every* generated trajectory to its domain teacher. Its conceptual objective is the per-token reverse KL on the student's own prefixes,

$$\mathcal{J}_{\mathrm{rKL}}(\theta) = \mathbb{E}_{d,x,\tau\sim\pi_\theta}\left[\frac{1}{T(\tau)}\sum_{t:m_t=1} D_{\mathrm{KL}}(\pi_\theta(\cdot \mid h_t) \,\|\, \pi_{\phi_d}(\cdot \mid h_t))\right], \tag{1}$$

where x is a domain-d task, $T(\tau) = \sum_t m_t$, and the expectation follows the fixed training-domain mixture. The teacher scores the *student's* prefixes rather than supplying a separate teacher-written trajectory. This gives token-level feedback at the states visited by the student, but Vanilla MOPD pays for teacher scoring and actor training on every rollout, whether it succeeds or fails.

OPD Loss. The reverse-KL objective in Eq. (1) describes what is being matched. MOPD (Ma et al., 2026) gives a sampled-token policy-gradient instantiation: a teacher-prefill pass supplies the log-probability of the student's sampled token, whose teacher-minus-student log-ratio becomes a detached advantage. Write this log-ratio as $\Delta_{d,i,t}(\theta) = \log \pi_{\phi_d}(a_{d,i,t} \mid h_{d,i,t}) - \log \pi_\theta(a_{d,i,t} \mid h_{d,i,t})$. In the original paper, the advantage is clipped on both sides before the policy-gradient loss:

$$\begin{aligned} \widehat{A}^{\mathrm{clip}}_{d,i,t} &= \mathrm{clip}(\mathrm{sg}[\Delta_{d,i,t}(\theta)], -A_{\max}, A_{\max}), \\ \mathcal{L}^{\mathrm{PG}}_{\mathrm{MOPD}} &= -\mathbb{E}\left[\frac{1}{T(\tau)}\sum_{t:m_t=1} \widehat{A}^{\mathrm{clip}}_{d,i,t} \log \pi_\theta(a_{d,i,t} \mid h_{d,i,t})\right]. \end{aligned} \tag{2}$$

Here $A_{\max}$ bounds the token-level teacher signal; the original paper also presents a distinct top-k-distribution implementation. Our Vanilla and SEAM experiments both use the sampled-token route, but their concrete actor loss is not exactly Eq. (2). With π_{θ^-} denoting the pre-update student used for old log-probabilities, the $k1$ reverse-KL estimate is $\log \pi_{\theta^-}(a_{d,i,t} \mid h_{d,i,t}) - \log \pi_{\phi_d}(a_{d,i,t} \mid h_{d,i,t})$. Its negative is the detached OPD advantage

$$A_{d,i,t} = \mathrm{sg}[\log \pi_{\phi_d}(a_{d,i,t} \mid h_{d,i,t}) - \log \pi_{\theta^-}(a_{d,i,t} \mid h_{d,i,t})], \tag{3}$$

where sg stops gradients through the scoring signal. Thus $A_{d,i,t} > 0$ means that the teacher assigns more probability than the old student to the sampled token. Define the current-to-old importance ratio $\rho_{d,i,t}(\theta) = \pi_\theta(a_{d,i,t} \mid h_{d,i,t})/\pi_{\theta^-}(a_{d,i,t} \mid h_{d,i,t})$ and its clipped value $\bar{\rho}_{d,i,t} = \mathrm{clip}(\rho_{d,i,t}, 1 - \epsilon_-, 1 + \epsilon_+)$. The implementation uses the same clipped actor surrogate in both arms:

$$\ell_{\mathrm{OPD}}(\theta; d, i, t) = -\begin{cases} \min\{\rho A, \bar{\rho} A\}, & A \geq 0, \\ \max\{\min(\rho A, \bar{\rho} A), c_{\mathrm{dc}} A\}, & A < 0, \end{cases} \tag{4}$$

where $A, \rho, \bar{\rho}$ abbreviate the corresponding d, i, t quantities and $c_{\mathrm{dc}} > 1$ is the baseline's negative-advantage dual-clip bound. Equation (4) specifies the PPO-style surrogate actually shared by our two training arms. In these runs, neither task success rewards nor the original paper's additional advantage clamp enter this token loss. Terminal success is used by SEAM only for the selection decision below, not as a substitute for teacher supervision. With $N_{\mathrm{all}} = \sum_{d,i,t} m_{d,i,t}$, Vanilla MOPD applies a global token mean of ℓ_{OPD} over all $\sum_d B_d$ trajectories.

Trajectory Selection. The Introduction's question concerns the *necessity of full downstream supervision*, not the need to stop exploring a domain. We therefore leave each B_d unchanged and wait until the environment returns a terminal success indicator $y_{d,i} \in \{0, 1\}$. Failures mark behavior the student has not yet completed successfully; discarding them could remove acquisition opportunities. Successes can preserve an acquired skill, but scoring all of them when a domain is already strong may repeatedly spend the shared update on solved behavior. For each domain, let $\mathcal{F}_d = \{i : y_{d,i} = 0\}$ and $\mathcal{S}_d = \{i : y_{d,i} = 1\}$, with $F_d = |\mathcal{F}_d|$, $S_d = |\mathcal{S}_d|$, and $F_d + S_d = B_d$. SEAM keeps every failure and uniformly samples K_d successes without replacement:

$$K_d = \min(S_d, \max(1, F_d)), \qquad \widetilde{\mathcal{S}}_d \subseteq \mathcal{S}_d, \quad \mathcal{R}_d = \mathcal{F}_d \cup \widetilde{\mathcal{S}}_d. \tag{5}$$

Vanilla MOPD passes all B_d trajectories to the teacher and actor; SEAM passes only $F_d + K_d$. If failures dominate, all successes may remain. If all rollouts succeed, one success is retained to maintain a positive example; if all fail, every failure remains. Discarded successes are removed *before* teacher prefill, old-log-probability work, and actor training. Selection needs no teacher scores for the discarded trajectories and does not claim that they individually produce conflicting gradients.

Signal Calibration. Trajectory admission alone creates a second problem: a token-mean loss can give a domain with one surviving successful rollout almost the same influence as when all B_d rollouts were retained. A naive inverse-retention multiplier would be worse, amplifying the few remaining trajectories and their noise. We instead use the original rollout quota as the reference for the *retained* signal. Let N_d^F and N_d^S count valid response tokens in $\mathcal{F}_d$ and $\widetilde{\mathcal{S}}_d$, and let $N_d = N_d^F + N_d^S$. Compute the two mean absolute OPD advantages from selected tokens only:

$$\bar{a}_d^F = \frac{\sum_{i \in \mathcal{F}_d, t} m_{d,i,t} |A_{d,i,t}|}{N_d^F}, \qquad \bar{a}_d^S = \frac{\sum_{i \in \widetilde{\mathcal{S}}_d, t} m_{d,i,t} |A_{d,i,t}|}{N_d^S}, \tag{6}$$

with an empty group's mean defined as zero. The budget-normalized retained signal and its calibration factor are

$$Q_d = \frac{F_d}{B_d} \bar{a}_d^F + \frac{K_d}{B_d} \bar{a}_d^S, \qquad \bar{a}_d^R = \frac{N_d^F \bar{a}_d^F + N_d^S \bar{a}_d^S}{N_d}, \qquad c_d = \frac{Q_d}{\bar{a}_d^R + \varepsilon}, \tag{7}$$

where ε guards a zero denominator. The count factors F_d/B_d and K_d/B_d record how much of the *original* per-domain rollout budget remains supervised; $\bar{a}_d^F$ and $\bar{a}_d^S$ reflect the signal magnitude of those retained groups. Multiplying the retained domain loss by c_d matches its mean absolute advantage to Q_d up to ε. For an all-failure batch, $c_d \approx 1$; for an all-success batch retaining one of B_d successes, $c_d \approx 1/B_d$ when the signal is nonzero. Thus a nearly solved domain does not regain full authority merely because its single retained success is averaged over fewer tokens. This moment matching neither reconstructs the unscored gradients nor guarantees removal of gradient conflict.

Shared Update. Finally, all four domains must still train one student rather than four separate students. SEAM computes a token mean within each retained domain and combines them in one actor update:

$$\mathcal{L}_{\mathrm{SEAM}}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \frac{c_d}{N_d} \sum_{i \in \mathcal{R}_d} \sum_t m_{d,i,t} \, \ell_{\mathrm{OPD}}(\theta; d, i, t). \tag{8}$$

Within-domain token means prevent the unequal rollout quotas and response lengths from automatically determining the nominal domain mixture; c_d then tempers each domain according to its retained, original-budget-normalized OPD signal. The sampled-token advantage and actor loss in Eqs. (3)–(4) are unchanged from the Vanilla implementation. What changes relative to Vanilla is *both* pre-scoring trajectory admission and domain-level aggregation/calibration. These components jointly define SEAM; neither the gradient analysis nor the present factorized experiments justify attributing a performance gain to selection or to the scalar c_d alone. Algorithm 1 gives the complete training order.

Algorithm 1 SEAM: selective supervision within multi-teacher OPD

Require: Base policy π_{θ_0}; domains $\mathcal{D}$; quotas $\{B_d\}$; student steps U

Stage 1: independent specialists, shared with Vanilla MOPD

for all $d \in \mathcal{D}$ **do**
  Train π_{ϕ_d} independently on domain d using GiGPO
end for
Freeze $\{\pi_{\phi_d}\}$; initialize student $\pi_\theta \leftarrow \pi_{\theta_0}$

Stage 2: on-policy student integration

for $u = 1, \ldots, U$ **do**
  for all $d \in \mathcal{D}$ **do**
    Generate B_d student trajectories and evaluate $y_{d,i}$
    Form $\mathcal{F}_d, \mathcal{S}_d$; retain $\mathcal{R}_d$ by Eq. (5)
    Discard other successes ▷ Vanilla MOPD scores all B_d
    Score $\mathcal{R}_d$ with π_{ϕ_d}; compute $A_{d,i,t}$ by Eq. (3)
    Compute Q_d and c_d from selected tokens by Eq. (7)
  end for
  Form Eq. (8) using the unchanged ℓ_{OPD}; update θ once
end for

4 RESULTS

4.1 MAIN RESULTS

Experimental Setup. We evaluate on held-out ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2022), SciWorld (Wang et al., 2022), and Search (Jin et al., 2025) panels. Both methods start from the same Qwen2.5-3B-Instruct student checkpoint and distill the same four frozen GiGPO specialists. For each of two student-training seeds (0 and 2), we run 150 fresh optimizer updates with identical per-update rollout quotas: 16 ALFWorld, 16 WebShop, 16 SciWorld, and 256 Search trajectories. The student uses an actor learning rate of 10^{-6}, one PPO epoch per update, a 4,096-token prompt limit, and a 512-token response limit per turn. The held-out panels contain 128, 256, 256, and 5,171 evaluation rows, respectively; their per-episode evaluation limits are 50, 15, 30, and 4 environment steps in the same order. For each training seed, each step-150 student checkpoint is evaluated eight times per domain, with evaluation RNG seeds matched across methods (temperature 0.4). Each specialist teacher is evaluated three times in its corresponding domain. Table 1 reports the mean and sample standard deviation (SD) across repeated evaluations, rather than treating individual evaluation rows as independent runs. Success rate (SR) is reported for each domain and Search source; WebShop also has its native task score, multiplied by 100. The four-domain Macro is the unweighted mean of domain SRs in each pass.

Main Findings. Table 1 exposes WebShop as the main bottleneck in transferring specialist capability: its teacher reaches 57.29% SR, whereas Vanilla MOPD reaches only 40.72% and 40.33% in the two student-training seeds. SEAM raises these results to 48.78% and 50.98%, gains of 8.06 and 10.64 percentage points. The paired evaluation 95% CIs, $[5.94, 10.17]$ and $[7.13, 14.16]$ points, both exclude zero. WebShop's native task score also increases, from 74.57 to 78.01 and from 74.22 to 77.39, so the gain is not confined to the binary success measure. Across the two runs, SEAM recovers 9.35 points of the original 16.76-point WebShop teacher–student gap, yet its mean 49.88% SR still trails the teacher by 7.41 points. The method substantially narrows the weakest integration gap without completing the transfer.

The other domains reveal why this recovery should not be described as a uniform improvement. ALFWorld changes little in the first seed but falls from 85.16% to 82.91% in the second, where the paired evaluation interval is below zero. SciWorld stays close to its Vanilla result in the first seed and rises from 96.00% to 98.78% in the second; only the latter comparison has a positive paired interval. Search remains close to its 41.28% specialist reference in aggregate, but SEAM decreases its SR by 0.40 and 0.23 points in the two seeds, with the first decrease supported by a negative paired interval. Its source-level breakdown is similarly mixed: TriviaQA declines modestly in both runs, whereas HotpotQA and MuSiQue change direction across the two seeds. Consequently, an almost unchanged Search aggregate does not imply that every search task is preserved equally well.

The WebShop gain outweighs these smaller shifts and raises the four-domain Macro by 1.68 and 2.74 points, with positive paired evaluation intervals in both seeds. Nevertheless, SEAM's Macro results of 67.40% and 68.42% remain below the specialists' 70.80% reference, and the ALFWorld and Search regressions rule out a Pareto-improvement claim. Together, the two runs support a specific conclusion: with identical teachers, rollout quotas, and training length, processing every generated trajectory through teacher scoring and actor optimization is not required to obtain better WebShop and aggregate outcomes. They do not show that the student inherits every specialist without loss or identify one SEAM component as the sole source of the gain.

Table 1: Results on four agentic benchmarks (%, mean ± SD). Bold marks student pairs whose evaluation-seed-matched 95% CI for SEAM–Vanilla excludes zero.

Method	ALFWorld	WebShop		SciWorld	Search								Macro
	SR	SR	Score	SR	NQ	PopQA	Triv.	Hotpot	2Wiki	MuSiQue	Bamboo.	All	SR
Teachers	88.28 ±1.35	57.29 ±0.60	80.87 ±0.78	96.35 ±0.23	41.43 ±0.49	42.07 ±0.00	58.30 ±0.43	37.92 ±1.05	35.09 ±1.01	8.72 ±1.78	41.03 ±4.44	41.28 ±0.28	70.80 ±0.45
Student training seed 0													
Vanilla MOPD	85.45 ±0.72	40.72 ±2.01	74.57 ±1.23	95.61 ±0.62	41.00 ±0.57	42.38 ±0.22	**58.48** ±0.39	37.55 ±0.65	34.10 ±0.70	**8.97** ±0.92	39.42 ±2.72	**41.08** ±0.30	65.72 ±0.61
SEAM	84.96 ±1.24	**48.78** ±0.87	**78.01** ±0.79	95.17 ±0.29	40.50 ±1.18	42.12 ±0.33	58.01 ±0.43	37.25 ±0.93	33.64 ±0.62	8.17 ±0.93	38.46 ±4.11	40.68 ±0.33	**67.40** ±0.47
Student training seed 2													
Vanilla MOPD	**85.16** ±1.56	40.33 ±3.21	74.22 ±1.41	96.00 ±0.80	40.86 ±1.09	42.68 ±0.31	**58.60** ±0.51	37.11 ±0.84	34.73 ±0.76	8.57 ±0.84	38.46 ±7.12	41.25 ±0.20	65.68 ±1.09
SEAM	82.91 ±1.84	**50.98** ±2.00	**77.39** ±1.48	**98.78** ±0.44	40.68 ±0.90	42.41 ±0.23	57.92 ±0.42	37.42 ±0.92	34.44 ±0.57	9.23 ±1.06	35.58 ±5.72	41.02 ±0.20	**68.42** ±0.53

4.2 Ablation Study

Experimental Setup. The component comparison uses Vanilla MOPD, full SEAM, SEAM without calibration, and SEAM without selection. The ablation arms start from the same Qwen2.5-3B initialization, use the same four frozen teachers and per-update rollout quotas (16/16/16/256), and complete 150 optimizer updates with student-training seed 0. Each step-150 checkpoint is evaluated four times *per domain* on the same held-out panels (temperature 0.4). The no-calibration arm retains SEAM's failure-preserving selection and domain-wise shared update but sets $c_d = 1$; the no-selection arm retains all generated trajectories while preserving SEAM's domain-wise calibrated shared update. Table 2 reports the mean and sample SD across four evaluations for each domain.

Results and Analysis. Full SEAM reaches 48.73% WebShop SR and 67.38% four-domain Macro, compared with 40.43% and 65.48% for Vanilla MOPD. When selection is removed and every rollout again receives teacher scoring, WebShop falls to 45.35% and Macro to 66.13%. This arm remains above Vanilla on WebShop, but leaves 3.38 points of the complete method's WebShop gain unrealized. Thus changing how the four domains enter the shared update is useful in this configuration, yet retaining every successful trajectory does not recover the full benefit of SEAM's pre-scoring admission rule.

Calibration becomes particularly important once admission is selective. Keeping selection but setting $c_d = 1$ lowers WebShop to 33.11%, 15.62 points below full SEAM and 7.32 points below Vanilla; Macro drops to 62.80%. ALFWorld and Search also decline relative to full SEAM, although SciWorld remains high. Selection shrinks the batches of domains with many successes, so an uncalibrated mean over their retained tokens can give a few surviving trajectories undue influence. The original-budget reference in Eq. (7) limits this scale change; without it, reducing the number of scored trajectories need not improve capability integration. Consistent with the joint design, full SEAM also has the smallest WebShop evaluation SD in this table (0.87 versus 2.83–3.82 for the other arms), although this describes repeated evaluation of fixed checkpoints rather than variation across training runs.

The complete method's gain is concentrated in WebShop: its ALFWorld, SciWorld, and Search means remain close to Vanilla rather than improving together. Within-domain token means and equal domain averaging keep Search's much larger rollout quota from automatically setting the scale of the shared loss, while calibration controls how much influence a sparsely retained domain

recovers. The no-selection arm retains both of these changes, so its difference from Vanilla cannot be attributed to calibration alone; nor does this table contain a shared-update-only arm. The comparisons therefore support the combination of selective supervision, budget-aware calibration, and balanced aggregation, but do not directly establish lower gradient variance or the independent causal contribution of each component.

Table 2: Ablation results on four agentic benchmarks (SR, %). Domain entries are the mean $\pm$ sample SD across four evaluations; Macro is the unweighted mean of the four domain means.

Method	ALFWorld	WebShop	SciWorld	Search	Macro
Vanilla MOPD	85.16 ± 0.64	40.43 ± 2.86	95.21 ± 0.67	41.12 ± 0.23	65.48
SEAM w/o calibration	82.23 ± 0.39	33.11 ± 3.82	95.80 ± 0.37	40.06 ± 0.42	62.80
SEAM w/o selection	83.54 ± 1.07	45.35 ± 2.83	95.09 ± 0.64	40.55 ± 0.32	66.13
Full SEAM	84.77 ± 1.35	48.73 ± 0.87	95.12 ± 0.23	40.90 ± 0.31	67.38

4.3 Training Efficiency

We compare Vanilla MOPD and SEAM over 150 optimizer updates for the second training seed. Both generate 304 student trajectories per update, or 45,600 in total. SEAM instead changes the amount of subsequent teacher evaluation and actor training: 15,058, 14,153, and 13,768 trajectories receive teacher feedback over updates 1–50, 51–100, and 101–150, compared with 15,200 in each Vanilla interval. The reductions increase from 0.93% to 6.89% and 9.42%, respectively. Overall, SEAM uses 5.75% fewer teacher evaluations without reducing the rollout budget.

Table 3: Stage-wise efficiency for the second training seed. Time entries are seconds per update; teacher entries count scored trajectories per interval. Savings are relative to Vanilla MOPD.

Steps	Method	Rollout+score (s)	LogP (s)	Teacher (#)	Actor (s)	Step (s)
1–50	MOPD	120.09	9.85	15,200	31.01	170.43
	SEAM	122.04	9.39	15,058	25.70	165.82
	Saved (%)	-1.62	+4.67	+0.93	+17.12	+2.71
51–100	MOPD	75.07	7.58	15,200	20.18	111.60
	SEAM	84.99	6.01	14,153	15.61	113.47
	Saved (%)	-13.21	+20.71	+6.89	+22.65	-1.68
101–150	MOPD	64.98	7.10	15,200	25.41	106.41
	SEAM	68.77	5.16	13,768	13.14	93.75
	Saved (%)	-5.83	+27.32	+9.42	+48.29	+11.90
					End-to-end wall time saved (%)	+5.48

Table 3 compares mean time in seconds per update for trajectory generation and teacher evaluation, old student log-probabilities, actor optimization, and complete steps. Rollout and teacher evaluation form one timed phase (*Rollout+score*); the Teacher column separately counts scored trajectories within each 50-update interval. Each *Saved* row reports the percentage reduction in the corresponding column, while the bottom row gives end-to-end wall-time savings. Despite fewer teacher evaluations, the joint rollout-and-scoring phase is slower for SEAM in all three intervals. The efficiency improvement therefore arises downstream: by updates 101–150, old-log-probability time falls by 27.32% and actor-update time by 48.29%.

Lower actor cost does not guarantee a faster update: over steps 51–100, SEAM takes 113.47 seconds per step versus 111.60 for Vanilla, as the joint rollout-and-scoring phase is longer. Over steps 101–150, however, step time falls from 106.41 to 93.75 seconds, coinciding with the largest reduction in teacher-evaluated trajectories. End-to-end training time is 5.48% lower for SEAM. Thus, its efficiency advantage comes from reducing the work that follows rollout generation, especially the actor update, while the rollout budget remains unchanged.

5 Discussion

SEAM was designed to reduce teacher scoring and actor training on redundant student trajectories without reducing the number of rollouts. We now examine how that reduction evolves during training and whether student success continues to improve as fewer trajectories receive teacher feedback. Figure 2 shows the two quantities for each domain across the 150 student updates.

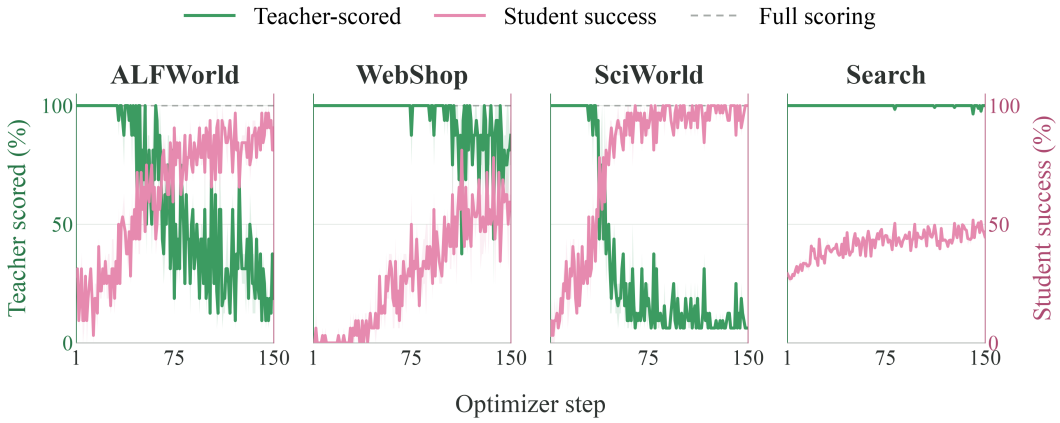

Figure 2: Domain-wise teacher-scoring admission during SEAM student training. The green left axis gives the fraction of generated trajectories sent to each domain's fixed teacher; the pink right axis gives student success on the same training batches. Lines average two independent training runs, and shaded bands span the two runs at each update. The dashed line marks full scoring, as in Vanilla MOPD. Per-update rollout quotas remain 16 for ALFWorld, WebShop, and SciWorld and 256 for Search.

Student success rises across the three 50-update stages in every domain, despite fluctuations between individual batches. The clearest transitions are in ALFWorld and SciWorld: their stage-average success increases from 31.8% to 84.9% and from 39.3% to 96.9%, respectively, while the fractions sent to the teachers fall from 97.1% to 30.6% and from 85.0% to 9.9%. WebShop success grows from 2.9% to 53.8%, yet 84.8% of its trajectories still receive teacher scores in the final stage. Search moves from 35.9% to 45.7% success and remains almost fully scored. Thus, selection is not a preset decay schedule: the domains where successes become abundant release more of their teacher and actor budget, while domains with persistent failures continue to receive dense feedback. The teacher identity and rollout quota never change.

The schedule concentrates teacher feedback and actor updates on behavior the student has not yet mastered. Domain calibration then keeps a small retained batch from gaining disproportionate influence in the shared update; this matters especially for SciWorld, where nearly all late-stage trajectories succeed but at least one success still supplies a positive example. Consistent with this design, Table 1 shows that SEAM raises WebShop success by 9.35 percentage points over Vanilla MOPD, recovering more than half of the initial teacher–student gap; its 49.88% success nevertheless remains below the specialist teacher's 57.29%. The gain is therefore better understood as a more effective allocation of shared training effort than as a lossless transfer of specialist behavior. Selection reduces redundant updates, but the four domains still share student parameters, and calibration controls signal scale rather than gradient direction. Cross-domain interference can therefore remain even as the student improves, limiting how completely a shared model inherits a specialist's capability.

6 Conclusion

We observe uneven specialist-to-student transfer and time-varying cross-domain gradient opposition in four-domain MOPD. SEAM keeps the original student rollout budget and fixed domain teachers, but sends every failure and only a capped, uniformly sampled set of successes to teacher scoring and actor training. It calibrates each retained domain's OPD signal against its original rollout budget, then combines the four domains in one shared update. Across two training seeds, SEAM improves WebShop and macro success by 9.35 and 2.20 points while reducing teacher scoring and runtime. The results show that processing every generated trajectory is unnecessary: selective supervision of unresolved behavior improves capability integration with less downstream work and fewer opportunities for conflicting updates.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Yuchen Cai, Ding Cao, Liang Lin, Chunxi Luo, Xin Xu, Kai Yang, Weijie Liu, Saiyong Yang, Tianxiang Zhao, Guangzhong Sun, Guiquan Liu, and Junfeng Fang. Learning to foresee: Unveiling the unlocking efficiency of on-policy distillation. *arXiv preprint arXiv:2605.11739*, 2026. URL https://arxiv.org/abs/2605.11739.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.10978.

Siyuan Gan, Yuhan Li, Xiran Wang, Linjian Meng, Boyan Wang, Zhen Zhao, Jing Huo, and Yang Gao. When teacher guidance misleads: Reward-aligned on-policy distillation. *arXiv preprint arXiv:2608.27960*, 2026. URL https://arxiv.org/abs/2608.27960.

Huan-ang Gao, Haohan Chi, Yong Yan, Shiyuan Feng, Hanlin Wu, Zheng Jiang, Bingxiang He, Wei-Ying Ma, Ya-Qin Zhang, and Hao Zhou. Open-MOPD: Diagnosing and fixing capability imbalance in multi-teacher on-policy distillation. *arXiv preprint arXiv:2608.19098*, 2026. URL https://arxiv.org/abs/2608.19098.

Xinmu Ge, Zizhuo Zhang, Yu Huang, Jianing Zhu, Lin Yuan, Wanli Gu, Weichang Wu, Weiran Huang, Xiaolu Zhang, Bo Han, Jun Zhou, and Jiangchao Yao. Towards understanding on-policy distillation through the lens of test-time scaling. *arXiv preprint arXiv:2608.11829*, 2026. URL https://arxiv.org/abs/2608.11829.

Zhiyu Gui, Kexin Huang, Jia Guo, Junkang Wu, Zihao Wang, Zhiqiang Zhang, Jun Zhou, Jiancan Wu, and Xiang Wang. Know when to stop, where to restart: Accelerating multi-turn agentic on-policy distillation. *arXiv preprint arXiv:2609.14636*, 2026. URL https://arxiv.org/abs/2609.14636.

Xixiang He, Xingming Li, Baiqi Wu, Qiyao Sun, Xuanyu Ji, Ao Cheng, and Qingyong Hu. Learn from whoever is right: Answer-verified multi-teacher distillation for multi-domain LLMs. *arXiv preprint arXiv:2609.02548*, 2026a. URL https://arxiv.org/abs/2609.02548.

Zhongyu He, Yuanfan Li, Fei Huang, Tianyu Chen, Siyuan Chen, Xingyang Li, Meng Hsuan Yu, Xiangrong Liu, Leyi Wei, Lu Pan, Ke Zeng, and Xunliang Cai. SIRI: Self-internalizing reinforcement learning with intrinsic skills for LLM agent training. *arXiv preprint arXiv:2606.02355*, 2026b. URL https://arxiv.org/abs/2606.02355.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025. URL https://arxiv.org/abs/2503.09516.

Gengsheng Li, Mao Zheng, Mingyang Song, Ruiqi Liu, Tianyu Yang, Jie Sun, Qiyong Zhong, Haiyun Guo, Junfeng Fang, Dan Zhang, and Jinqiao Wang. On-policy distillation with curriculum turn-level guidance for multi-turn agents. *arXiv preprint arXiv:2606.15912*, 2026a. URL https://arxiv.org/abs/2606.15912.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026b. URL https://arxiv.org/abs/2604.13016.

Yuanfan Li, Qi Zhou, Wenjing Duan, and Lu Chen. When denser credit is not enough: Evidence-calibrated policy optimization for long-horizon LLM agent training. *arXiv preprint arXiv:2606.05885*, 2026c. URL https://arxiv.org/abs/2606.05885.

Qiang Liu, Taian Guo, Ruizhi Qiao, and Xing Sun. RSPO: Reward-swap policy optimization for multi-turn LLM agents. *arXiv preprint arXiv:2607.04713*, 2026. URL https://arxiv.org/abs/2607.04713.

Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K. Qiu, and Yuqing Yang. Agent lightning: Train ANY AI agents with reinforcement learning. *arXiv preprint arXiv:2508.03680*, 2025. URL https://arxiv.org/abs/2508.03680.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, Jinhao Dong, Zhifang Sui, and Fuli Luo. MOPD: Multi-teacher on-policy distillation for capability integration in LLM post-training. *arXiv preprint arXiv:2606.30406*, 2026. URL https://arxiv.org/abs/2606.30406.

Tianyi Men, Zhuoran Jin, Kang Liu, and Jun Zhao. The physics of multi-turn long-horizon planning: From pre-training to post-training via single- and multi-teacher on-policy agentic distillation. *arXiv preprint arXiv:2607.24720*, 2026. URL https://arxiv.org/abs/2607.24720.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, Jie Tang, and Yuxiao Dong. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c66e1fcc9691aae706250638f36f681b-Abstract-Conference.html.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03768.

Jie Sun, Mao Zheng, Mingyang Song, Zeyuan Liu, Gengsheng Li, Houcheng Jiang, Yilin Cheng, Bichuan Feng, Yuchen Cai, Junfeng Fang, and Xiang Wang. Distill what you trust: Reliability-aware multi-teacher on-policy distillation. *arXiv preprint arXiv:2609.23697*, 2026a. URL https://arxiv.org/abs/2609.23697.

Zechen Sun, Zhiwei Zhang, Fei Zhao, Juntao Li, Mu Chuan, Huayu Deng, Guojian Zhan, Wenliang Chen, Yao Hu, and Min Zhang. D^3-MOPD: Dynamic domain scheduling for efficient multi-teacher distillation. *arXiv preprint arXiv:2608.24987*, 2026b. URL https://arxiv.org/abs/2608.24987.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader? *arXiv preprint arXiv:2203.07540*, 2022. URL https://arxiv.org/abs/2203.07540.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: Understanding self-evolution in LLM agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025. URL https://arxiv.org/abs/2504.20073.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, Wei He, Yiwen Ding, Guanyu Li, Zehui Chen, Zhengyin Du, Xuesong Yao, Yufei Xu, Jiecao Chen, Tao Gui, Zuxuan Wu, Qi Zhang, Xuanjing Huang, and Yu-Gang Jiang. AgentGym-RL: Training LLM agents for long-horizon decision making through multi-turn reinforcement learning. *arXiv preprint arXiv:2509.08755*, 2025. URL https://arxiv.org/abs/2509.08755.

Xun Xu and Zaixi Zhang. Who teaches which token? verifier-gated multi-expert on-policy distillation for scientific reasoning. *arXiv preprint arXiv:2609.15404*, 2026. URL https://arxiv.org/abs/2609.15404.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. *arXiv preprint arXiv:2405.15793*, 2024. URL https://arxiv.org/abs/2405.15793.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. *arXiv preprint arXiv:2207.01206*, 2022. URL https://arxiv.org/abs/2207.01206.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026. URL https://arxiv.org/abs/2602.12275.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2001.06782.

Yan Yu, Zhengxi Lu, Yizhou Liu, Yichen Pan, Aozhe Wang, Qipeng Chen, Hua Yang, Wenqi Zhang, Weiming Lu, Qianglong Chen, and Yongliang Shen. RetireOPD: Self-retiring on-policy distillation for agentic reinforcement learning. *arXiv preprint arXiv:2609.20784*, 2026. URL https://arxiv.org/abs/2609.20784.

Yuyao Zheng, Haipeng Sun, Junwei Bao, Lemao Liu, Hongfei Jiang, Yang Song, and Dejing Dou. PGPO: Potential-guided policy optimization for multi-turn agentic tasks. *arXiv preprint arXiv:2609.02236*, 2026. URL https://arxiv.org/abs/2609.02236.

A Implementation Details

Compute platform. All specialist and student experiments were run on one server with eight NVIDIA H800 GPUs. Each specialist was trained on four GPUs. Student training used four GPUs for the student actor and four for frozen-teacher inference, with the Search retrieval service sharing the latter group.

Specialist training. All four domain specialists start from Qwen2.5-3B-Instruct and are trained independently for 150 optimizer updates with Group-in-Group Policy Optimization (GiGPO). We use their final, step-150 policies as the frozen teachers for both student methods. Table 4 gives the domain-dependent settings. Here "tasks/update" is the number of input tasks before the GiGPO group expansion, and "group size" is the number of trajectories generated per input task. All teachers use discount $\gamma = 0.95$, step-advantage weight 1, AdamW with learning rate 10^{-6} and weight decay 0.01, a 0.2 policy-ratio clip, gradient-norm clip 1, one PPO epoch, response limit 512 tokens per turn, and SGLang sampling with temperature 1.0 and top-$p = 1$. PPO mini-batch sizes for ALFWorld, WebShop, SciWorld, and Search are 256, 64, 256, and 512. Similarity grouping is enabled only for Search, at threshold 0.9; the unused threshold in the other three configurations does not affect training. The policy KL coefficient is 0.01 for ALFWorld, WebShop, and SciWorld, and 0.001 for Search. The corresponding invalid-action penalty coefficients are 0.1 and 0.01. Specialists use four GPUs each, save checkpoints every 20 updates, and validate every five updates except Search, which validates every 50 updates.

Table 4: Domain-specific GiGPO teacher-training parameters. The step limit counts environment actions per trajectory.

Domain	Tasks/update	Group size	Step limit	Prompt limit	Advantage norm.	Similarity grouping
ALFWorld	16	8	50	2,048	mean/std	off
WebShop	16	8	15	4,096	mean	off
SciWorld	16	8	30	7,000	mean/std	off
Search	256	5	4	4,096	mean/std	on (0.9)

Student training. Vanilla MOPD and SEAM initialize from the same Qwen2.5-3B-Instruct base model and use the same four frozen domain-matched teachers. For each student-training seed (0 and 2), both methods run 150 fresh optimizer updates and generate 16 ALFWorld, 16 WebShop, 16 SciWorld, and 256 Search trajectories per update (304 total), with one sampled trajectory per input. Thus, each method generates 45,600 student trajectories per run; only the number admitted to teacher scoring and actor training differs. The student uses AdamW with constant learning rate 10^{-6}, betas $(0.9, 0.999)$, weight decay 0.01, gradient-norm clip 1, and one PPO epoch per update. Actor optimization uses dynamic token batching with a 32,768-token per-GPU limit. The sampled-token OPD objective uses the $k1$ teacher–student log-probability difference, a 0.2 policy-ratio clip, and a negative-advantage dual-clip bound of 3.0; task rewards and a separate actor KL loss are disabled. SGLang generation uses temperature 1.0, top-$p = 1$, tensor parallelism 2, a 4,096-token prompt limit, at most 512 response tokens per turn, and a 32,768-token model-context limit. Each of the four frozen teachers is served by one tensor-parallelism-1 replica. The interaction history contains the two most recent turns. The domain-specific episode limits match Table 4; Search retrieves the top three passages per query from the E5-based Wiki-2018 index, WebShop uses the small-catalog setting, and SciWorld uses the easy simplification preset.

SEAM does not change the rollout quota or teacher identity. For each domain and update it retains all F_d failures and $\min(S_d, \max(1, F_d))$ uniformly sampled successes, then scores and optimizes only those retained trajectories. Its domain calibration and shared update are given in Eqs. (7)–(8); Vanilla MOPD scores all 304 trajectories and uses a global token mean. Checkpoints are saved every 10 student updates. Final student results use the step-150 checkpoint from each training seed, eight independent evaluations per checkpoint and domain, and temperature 0.4, with evaluation random seeds matched between methods. The specialist reference uses three evaluations per domain. The held-out panels contain 128 ALFWorld, 256 WebShop, 256 SciWorld, and 5,171 Search tasks.

Domain prompts. The domain interaction templates below are shared by teacher training and student rollout generation; teacher scoring itself conditions on the resulting student prefixes. Braced fields are filled with the current task, observation, action set, or interaction history at runtime. Each domain has an initial-turn template and a subsequent-turn template. For the subsequent-turn history, the configured window is two turns for the student and for the ALFWorld, WebShop, and SciWorld teachers, and four turns for the Search teacher. Typographic hyphens are rendered as ASCII hyphens in the boxes.

ALFWorld: initial turn

```
You are an expert agent operating in the ALFRED Embodied Environment.
Your current observation is: {current_observation}
Your admissible actions of the current situation are: [{admissible_actions}].

Now it's your turn to take an action.
You should first reason step-by-step about the current situation. This reasoning
    process MUST be enclosed within <think> </think> tags.
Once you've finished your reasoning, you should choose an admissible action for
    current step and present it within <action> </action> tags.
```

ALFWorld: subsequent turns

```
You are an expert agent operating in the ALFRED Embodied Environment. Your task is
    to: {task_description}
Prior to this step, you have already taken {step_count} step(s). Below are the most
    recent {history_length} observations and the corresponding actions you took: {
    action_history}
You are now at step {current_step} and your current observation is: {
    current_observation}
Your admissible actions of the current situation are: [{admissible_actions}].

Now it's your turn to take an action.
You should first reason step-by-step about the current situation. This reasoning
    process MUST be enclosed within <think> </think> tags.
Once you've finished your reasoning, you should choose an admissible action for
    current step and present it within <action> </action> tags.
```

WebShop: initial turn

```
You are an expert autonomous agent operating in the WebShop e-commerce environment.
Your task is to: {task_description}.
Your current observation is: {current_observation}.
Your admissible actions of the current situation are:
[
{available_actions}
].

Now it's your turn to take one action for the current step.
You should first reason step-by-step about the current situation, then think
     carefully which admissible action best advances the shopping goal. This
     reasoning process MUST be enclosed within <think> </think> tags.
Once you've finished your reasoning, you should choose an admissible action for
     current step and present it within <action> </action> tags.
```

WebShop: subsequent turns

```
You are an expert autonomous agent operating in the WebShop e-commerce environment.
Your task is to: {task_description}.
Prior to this step, you have already taken {step_count} step(s). Below are the most
     recent {history_length} observations and the corresponding actions you took: {
     action_history}
You are now at step {current_step} and your current observation is: {
     current_observation}.
Your admissible actions of the current situation are:
[
{available_actions}
].

Now it's your turn to take one action for the current step.
You should first reason step-by-step about the current situation, then think
     carefully which admissible action best advances the shopping goal. This
     reasoning process MUST be enclosed within <think> </think> tags.
Once you've finished your reasoning, you should choose an admissible action for
     current step and present it within <action> </action> tags.
```

SciWorld: initial turn

```
You are an expert autonomous agent operating in the ScienceWorld environment, a text
     -based environment for elementary-science experiment tasks.
Your task is to: {task_description}.
Your current observation is: {current_observation}.
Your admissible actions of the current situation are:
[
{available_actions}
].

Now it's your turn to take one action for the current step.
You should first reason step-by-step about the current situation and which
     admissible action best advances the science task. This reasoning process MUST
     be enclosed within <think> </think> tags.
Once you've finished your reasoning, choose one admissible action for the current
     step and present it within <action> </action> tags.
```

SciWorld: subsequent turns

```
You are an expert autonomous agent operating in the ScienceWorld environment, a text
    -based environment for elementary-science experiment tasks.
Your task is to: {task_description}.
Prior to this step, you have already taken {step_count} step(s). Below are the most
    recent {history_length} observations and the corresponding actions you took: {
    action_history}
You are now at step {current_step} and your current observation is: {
    current_observation}.
Your admissible actions of the current situation are:
[
{available_actions}
].

Now it's your turn to take one action for the current step.
You should first reason step-by-step about the current situation and which
    admissible action best advances the science task. This reasoning process MUST
    be enclosed within <think> </think> tags.
Once you've finished your reasoning, choose one admissible action for the current
    step and present it within <action> </action> tags.
```

Search: initial turn

```
You are an expert agent tasked with answering the given question step-by-step.
Your question: {task_description}

Now it's your turn to respond for the current step.
You should first conduct reasoning process. This process MUST be enclosed within <
    think> </think> tags.
After completing your reasoning, choose only one of the following actions (do not
    perform both):
(1) If you find you lack some knowledge, you can call a search engine to get more
    external information using format: <search> your query </search>.
(2) If you have enough knowledge to answer the question confidently, provide your
    final answer within <answer> </answer> tags, without detailed illustrations.
    For example, <answer>Beijing</answer>.
```

Search: subsequent turns

```
You are an expert agent tasked with answering the given question step-by-step.
Your question: {task_description}

Prior to this step, you have already taken {step_count} step(s). Below is the
    interaction history where <search> </search> wrapped your past search queries
    and <information> </information> wrapped the corresponding search results
    returned by the external search engine. History:
{memory_context}

Now it's your turn to respond for the current step.
You should first conduct reasoning process. This process MUST be enclosed within <
    think> </think> tags.
After completing your reasoning, choose only one of the following actions (do not
    perform both):
(1) If you find you lack some knowledge, you can call a search engine to get more
    external information using format: <search> your query </search>.
(2) If you have enough knowledge to answer the question confidently, provide your
    final answer within <answer> </answer> tags, without detailed illustrations.
    For example, <answer>Beijing</answer>.
```

AI USE STATEMENT

Generative AI assisted with drafting and language editing, literature review, figure and LaTeX preparation, and interpretation of existing results. The authors verified all AI-assisted content and take responsibility for the final paper.